General Markov Modeling of Pop-Up Threats With Applications To Persistent Area Denial [1]

Yong Liu and Jose B. Cruz, Jr.
Department of Electrical and Computer Engineering,
The Ohio State University
2015 Neil Avenue, Columbus, OH 43210-1272

Corey J. Schumacher
AFRL/VACA
2210 8th Street, Wright Patterson AFB, OH 45433-7531

Abstract – Pop-up threats usually appear or disappear randomly in a battle field. If the next pop-up threat locations could be predicted it would assist a search or attack team, such as in a Persistent Area Denial (PAD) mission, in getting a team of unmanned air vehicles (UAVs) to the threats sooner. We present a Markov model for predicting pop-up ground threats in military operations. We first introduce a general Markov chain of order n to capture the dependence of the appearance of pop-up threats on previous locations of the pop-up threats over time. We then present an adaptive approach to estimate the stationary transition probabilities of the n^{th} order Markov models. To choose the order of the Markov chain model for a specific application, we also discuss hypothesis tests from statistical inference on historical data of pop-up threat locations. Anticipating intelligent responses from an adversary, which might change its pop-up threat deployment strategy upon observing UAV movements, we present adaptive Markov chain models using a moving horizon approach to estimate possibly abrupt changes in transition probabilities. We consider the problem of cooperative control among multiple networked UAVs for the PAD mission. The combined information of predicted and actual pop-up target locations is utilized to develop efficient cooperative strategies for networked UAVs. Both a theoretical analysis and simulation results are presented to evaluate the Markov model used for predicting pop-up threats. These results demonstrate the effectiveness of cooperative

[1] This work was supported by the AFRL/VA and AFOSR Collaborative Center of Control Science at The Ohio State University (Grant F33615-01-2-3154). The authors gratefully acknowledge the inputs of Prof. Kevin M. Passino and Dr. Andrew G. Sparks. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the US Air Force, or the U.S. Government.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2006		**00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE
General Markov Modeling of Pop-Up Threats with Applications to Persistent Area Denial

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory,Air Vehicles Directorate,Wright Patterson AFB,OH,45433

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**31**	

strategies using the combined information of threats and predicted threats in improving overall mission performance.

Index terms: Pop-up threats; Markov chain model; Model order test; Cooperative strategies.

I. INTRODUCTION

Ground threats in military operations, in general, are either fixed or mobile. These threats, whether fixed or mobile, might be observed and tracked, or they might be hidden (perhaps in caves or underground) and not yet deployed. In the case of hidden threats, they become recognized as threats only after they are observed and tracked. Some threats might become hidden or taken out of deployment. Taken as a time series or sequence, these threats that appear and disappear in the course of a mission or battle are called "pop up" threats. Pop-up threats might appear and/or disappear at random times. The locations of known threats are often identified at the beginning of the battle, while a pop-up threat is not always observable prior to the start of the operation. In many cases, the probability of appearance of a pop-up threat at a location might depend on the appearance of pop-up threats at various previous locations. We capture this dependence by utilizing Markov models. The most closely related work is introduced in [1], [2] and [3]. In these references, the authors model the location of pop-up threats as a first-order Markov chain. In this case, the future appearance of a pop-up threat is dependent only on the current pop-up threat and independent of any previous pop-up threats. These papers focus on investigating the properties of the first-order Markov chain model for pop-up threats with stationary transition probabilities. In complex situations, a higher-order Markov chain model might be more realistic. We introduce a higher-order Markov chain model for predicting pop-up threats. We consider homogeneous (stationary) Markov chains only. Furthermore, the deployment of pop-up threats by the adversary might be changed in response to recent actions by the other force. We assume that the change of pop-up deployment could result from either a change of

transition probability values or a change of order of the Markov chain process describing the deployment of the pop-up threats. The adaptive Markov chain model for pop-up threats with changing transition probabilities introduced in this paper is capable of changing the values of the transition probabilities and changing its order. With this adaptive model, we allow sudden changes in the transition probabilities such as step changes. While the present paper is mainly concerned with the problem of modeling pop-up threats as a general Markov process, a potential application associated with pop-up threats is considered here for illustration. This application is the Persistent Area Denial (PAD) mission for multiple unmanned air vehicles. The aim of multiple UAVs in the PAD operation is to provide persistent surveillance, tracking, or rapid engagement with strike assets, against pop-up threats at various ranges. Cooperation among multiple UAVs is a key capability for utilizing the full potential of such systems.

The topic of coordinating UAVs for a military operation has been addressed in the literature recently. In [4], the author modeled the information flow and communication constraints between UAVs and developed a cooperative strategy for path re-planning to deal with pop-up threats during the mission. The work in [4], however, does not consider that the pop-up threats might appear and disappear in a stochastic manner. Obviously, uncertainty introduced by pop-up threats presents additional theoretical and technical challenges in designing and implementing cooperative control strategies for multiple UAVs. Predicting the locations of pop-up threats is an urgent task for UAV control in such time-critical missions. In [1-2], using a first-order Markov chain to model pop-up threats, the authors developed a cooperative control strategy for multiple UAVs in PAD operations via a mixed integer programming approach. In this strategy, the trajectories of vehicles are optimized so that they might reach all threats in the shortest time. In those papers the predicted information, however, is not used in the computation of initial cooperative control strategies. Only after all known threats are reached do the UAVs get assigned to predicted pop-up threats. In this paper, this

cooperative strategy is modified in a manner that allows the UAVs to make use of both information about actual locations of threats and information about predicted locations of pop-up threats, in coordinating their target assignments [3]. Moreover, since recent advances in telecommunication have provided enabling technologies for achieving cooperative control of multiple UAVs via communication networks [5], the networked UAV systems performing PAD operations are also presented in [3]. We investigate the performance of cooperative strategies using combined information with pop-up threats modeled as an adaptive Markov chain whose order may switch from first order to a higher order, or later switch to another order, and/or with transition probabilities that can change with the data, extending previous work [1-3] on first-order Markov modeling .

In the remainder of this paper, we present a general Markov-model of order n ($n \geq 1$) for pop-up threats, and develop an approach for generating Markov chain sequences of pop-up locations as a data base for computer simulation. We introduce an adaptive approach to estimate stationary transition probabilities and develop a moving-horizon approach to estimate changing (piece-wise stationary) transition probabilities. In section IV, we consider coordinating multiple UAVs for a PAD mission to illustrate the effectiveness of Markov models in predicting the locations of pop-up threats. We briefly review strategies for coordinating multiple UAVs using combined information. We also present Monte Carlo simulation examples performed on an Air Force Research Laboratory (AFRL) simulator to illustrate the performance of UAV cooperative strategies for pop-up appearance modeled by higher-order Markov chains and modeled by Markov chains with changing transition probabilities. The last section presents some concluding remarks.

II. MARKOV MODEL OF POP-UP THREATS

Pop-up threats are ground assets that appear at unpredictable locations and at random instants of time. In addition, pop-up threats may stay for a limited time interval and then disappear. In some cases, the location of a pop-up threat depends on a previous location of a pop-up target in the local

region. Such a discrete-time random sequence of the appearance of pop-up targets can be modeled as a Markov chain [6]. The region of a military operation is assumed to be a two-dimensional rectangular grid, with length L and width W. Divide the edge with length L into m segments and the edge with width W into n segments which partitions the entire area into mn smaller rectangles, each with a size of $lw = (L/m)(W/n)$. We number these cells so that the discretized state space is

$$Q = \{1, 2, \cdots, N_Q\}$$

where $N_Q = mn$. For example, the region considered in this paper is shown in Figure 1 with a length of $L = 5$ km and a width of $W = 5$ km. Let $m = n = 10$ so that the entire area is divided into 100 cells. We assume that there exists at most one target in each cell at one time and no target appears between cells of the partitioned area.

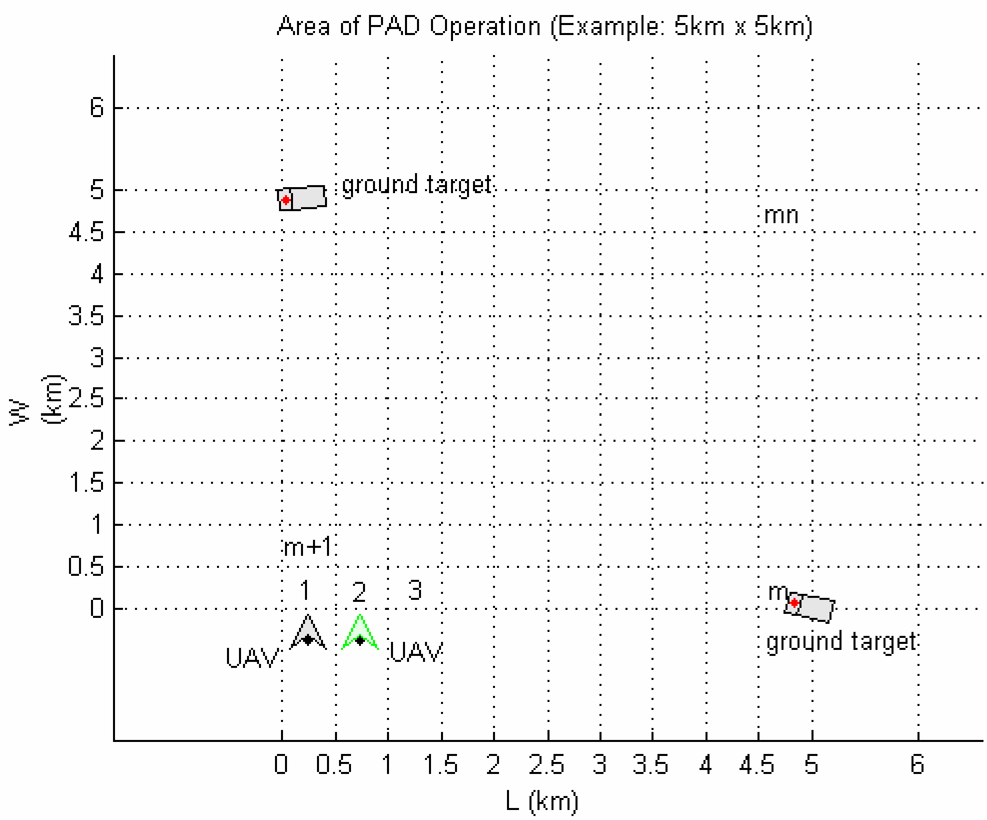

Figure 1. Theater of PAD operations

Let $Z_k \in Q$ be the cell at which the pop-up threat appears at discrete-event time index k. Note that the index k represents the time when a pop-up threat shows up. Let $Z = \{Z_k, k \geq 0\}$ represent a random sequence of locations of different pop-up threats.

A. First-order Markov-Chain Model

If the location of the next pop-up threat only depends on the location of the present threat, we model the predicted pop-up locations as a first-order Markov-chain such that, for any k, Z_{k+1} is independent of $Z_0, Z_1, \cdots, Z_{k-1}$ given Z_k. That is,

$$\Pr\left(Z_{k+1} = j \mid Z_k = i, Z_{k-1} = i_{k-1}, \cdots, Z_0 = i_0\right) = \Pr\left(Z_{k+1} = j \mid Z_k = i\right), \qquad i, j \in Q \ . \qquad (1)$$

In the expression above, $Z_k = i$ means that the pop-up target appears in cell i at discrete event index k. The implication is that the dependence structure is such that, given the location of a current pop-up threat, the next pop-up location can be characterized probabilistically without any additional information. We also assume that the one-step transition probabilities $\left\{\Pr\left(Z_{k+1} = j \mid Z_k = i\right), i, j \in Q\right\}$ depend only on the states i and j and are independent of the index k, i.e.,

$$\Pr\left(Z_{k+1} = j \mid Z_k = i\right) = p_{ij} \qquad \text{for all } i, j \in Q . \qquad (2)$$

This is said to be stationary or homogeneous. Thus, we are only concerned with homogeneous Markov chains. Let

$$P^{(1)} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N_Q} \\ p_{21} & p_{22} & \cdots & p_{2N_Q} \\ \cdots & \cdots & \cdots & \cdots \\ p_{N_Q1} & p_{N_Q2} & \cdots & p_{N_QN_Q} \end{bmatrix}$$

be the one-step transition probability matrix of the first-order Markov chain Z, where for any $i, j \in Q$, $p_{ij} \geq 0$ and for each $i \in Q$, $\sum_{j \in Q} p_{ij} = 1$.

B. Higher-order Markov-Chain Model

The dependence structure of order one is restrictive and may not sufficiently model some situations. For a more realistic model, it is assumed that future pop-up locations might depend on the state of the system during the previous pop-up threat events. This structure may be modeled as a Markov chain of order higher than one. For example, the future pop-up locations could depend on the previous n locations of the pop-up threats. This can be described as follows:

$$\begin{aligned}&\Pr\left(Z_{k+1} = j \mid Z_k = i_1, Z_{k-1} = i_2, \cdots, Z_{k-n+1} = i_n, \cdots, Z_0 = i_0\right) \\ &\qquad = \Pr\left(Z_{k+1} = j \mid Z_k = i_1, Z_{k-1} = i_2, \cdots, Z_{k-n+1} = i_n\right), \quad i_0, \cdots, i_n, j \in Q\end{aligned} \tag{3}$$

We consider a second-order Markov chain model in this paper. Higher order models can be developed similarly. In terms of the previous notations, the second-order dependence is characterized as

$$\Pr\left(Z_{k+1} = l \mid Z_k = j, Z_{k-1} = i, \cdots, Z_0 = i_0\right) = \Pr\left(Z_{k+1} = l \mid Z_k = j, Z_{k-1} = i\right), \quad i, j, l \in Q \tag{4}$$

The one-step transition probability matrix for the Markov chain Z of order two is given by

$$P^{(2)} = \begin{bmatrix} p_{111} & p_{112} & \cdots & p_{11N_Q} \\ p_{121} & p_{122} & \cdots & p_{12N_Q} \\ \cdots & \cdots & \cdots & \cdots \\ p_{N_Q N_Q 1} & p_{N_Q N_Q 2} & \cdots & p_{N_Q N_Q N_Q} \end{bmatrix} \tag{5}$$

where for any $i, j, l \in Q$, $p_{ijl} = \Pr\left(x_{k+1} = l \middle| x_k = j, x_{k-1} = i\right) > 0$ and for each $i, j, l \in Q$, $\sum_{l \in Q} p_{ijl} = 1$.

C. Computer Simulation of Pop-up Threats

The next step is to develop a procedure to generate pseudo-data for constructing the models and for evaluating the measures of mission performance and effectiveness. We simulate a sequence of locations of pop-up threats which is modeled as a Markov chain. Let $(\hat{Z}_0, \hat{Z}_1, \cdots)$ represent the

simulated sequence of cells in which pop-up threats are located within a given state space Q. For the first-order Markov chain model, let $p^{(0)} = \left(p_1, \cdots, p_i, \cdots, p_{N_Q} \right)$ represent a given initial distribution where p_i is the absolute initial probability of the pop-up threat appearing at cell i. Given a one-step transition probability matrix $P^{(1)}$, a sequence is generated by setting

$$\begin{aligned}
\hat{Z}_0 &= \psi_p^{(1)}(U_0) \\
\hat{Z}_1 &= \phi^{(1)}(\hat{Z}_0, U_1) \\
\hat{Z}_2 &= \phi^{(1)}(\hat{Z}_1, U_2) \\
&\vdots \\
\hat{Z}_L &= \phi^{(1)}(\hat{Z}_{L-1}, U_L) \quad L \geq 1
\end{aligned} \tag{6}$$

where $\{U_0, U_1, \cdots\}$ is a sequence of independent, identically distributed random variables, that is uniformly distributed on the unit interval $[0,1]$ and L is the length of the sequence to be generated and stored as test data. The key components in this procedure are two functions, which we call the initiation function and the update function. The initiation function $\psi_p^{(1)} : [0,1] \to Q$ is defined as

$$\psi_p^{(1)}(x) = \begin{cases} 1 & \text{for } x \in \left[0, p_1\right) \\ \vdots & \\ i & \text{for } x \in \left[\sum_{j=1}^{i-1} p_j, \sum_{j=1}^{i} p_j \right) \\ \vdots & \\ N_Q & \text{for } x \in \left[\sum_{j=1}^{N_Q - 1} p_j, 1 \right] \end{cases} \tag{7}$$

and the update function $\phi^{(1)} : Q \times [0,1] \to Q$ is defined as

$$\phi^{(1)}(i,x)=\begin{cases}1 & \text{for } x\in\left[0,p_{i1}\right)\\ \vdots & \\ j & \text{for } x\in\left[\sum_{l=1}^{j-1}p_{il},\sum_{l=1}^{j}p_{il}\right)\\ \vdots & \\ N_Q & \text{for } x\in\left[\sum_{l=1}^{N_Q-1}p_{il},1\right]\end{cases}\tag{8}$$

We note that the initiation function is applied to generate the position of the first pop-up location $\hat{Z}_0$, and the update function is applied to generate $\hat{Z}_{n+1}$ from $\hat{Z}_n$ at any time index n. Thus, we can apply this procedure iteratively to produce the entire chain $\{\hat{Z}_0,\hat{Z}_1,\cdots\}$. It can be verified that the choices of $\psi_p^{(1)}$ and $\phi^{(1)}$ satisfy the properties of first-order Markov chains [6].

Similarly, given a second-order Markov chain with the initial distribution $p^{(0)}$ and a one-step transition probability matrix $P^{(2)}$, we develop a procedure to generate a sequence of cells $(\hat{Z}_0,\hat{Z}_1,\cdots)$. Considering the dependency structure for the second-order Markov chain, we define an initiation function to generate the first two samples $\hat{Z}_0$ and $\hat{Z}_1$. The initiation function $\psi_p^{(2)}:\left[0,1\right]\times\left[0,1\right]\to Q\times Q$, is defined as

$$\psi_p^{(2)}(x,y)=(i,j)\quad\text{if}\quad x\in\left[\sum_{l=1}^{i-1}p_l,\sum_{l=1}^{i}p_l\right)\text{ and }\ y\in\left[\sum_{l=1}^{j-1}p_l,\sum_{l=1}^{j}p_l\right)\tag{9}$$

and the update function $\phi^{(2)}:Q\times Q\times[0,1]\to Q$ is given by

$$\phi^{(2)}(i,j,x)=s\text{ if }\ x\in\left[\sum_{l=1}^{s-1}p_{ijl},\sum_{l=1}^{s}p_{ijl}\right)\tag{10}$$

For the update function above, it is clear that the next state will depend on the immediately preceding two states. The pop-up locations of order two are then simulated by setting

$$\begin{aligned} \left(\hat{Z}_0, \hat{Z}_1\right) &= \psi_p^{(2)}(U_0, U_1) \\ \hat{Z}_2 &= \phi^{(2)}(\hat{Z}_0, \hat{Z}_1, U_2) \\ \hat{Z}_3 &= \phi^{(2)}(\hat{Z}_1, \hat{Z}_2, U_3) \\ &\vdots \\ \hat{Z}_L &= \phi^{(2)}(\hat{Z}_{L-2}, \hat{Z}_{L-1}, U_L) \quad L \geq 2 \end{aligned} \tag{11}$$

The length of the sequence, *L*, defines the size and extent of the test data base. This procedure can be easily extended to the case of simulating a sequence of pop-up locations of Markov chain of order higher than two. It should be noted that as we increase the order, the size of the data base increases. In particular, the number of elements in the transition probability matrices will increase as the n^{th} power of the number of cells, where n is the order of the model.

III. ESTIMATION OF ONE-STEP TRANSITION PROBABILITIES

A. Stationary Transition Probabilities

An adaptive approach to estimate the stationary transition probability matrix for the first-order Markov chain model has been developed in [1, 2]. We briefly review this adaptive algorithm for the first-order model and develop an extension to a higher-order Markov chain model. Let $n_{ij}(k)$ represent the status of transition from the previously observed pop-up threat at location $i \in Q$ at time *k-1* to the current observed pop-up threat at cell $j \in Q$ from a data sequence Z at time k. That is,

$$n_{ij}(k) = \begin{cases} 1 & \text{if state } i \text{ at time } k-1 \text{ transitions to state } j \text{ at time } k \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

For the first-order Markov chain, the updated stationary transition probability $\hat{p}_{ij}(k)$ of the next pop up threat to appear at cell j given the current pop-up threat at cell i, for all $i, j \in Q$, at discrete event index k can be estimated by

$$\hat{p}_{ij}(k) = \frac{\sum_{h=1}^{k} n_{ij}(h)}{\sum_{s=1}^{N_Q}\sum_{h=1}^{k} n_{is}(h)} \tag{13}$$

This update procedure increases the estimated stationary transition probability from a location *i* to a location j where a pop up has just appeared and reduces it for the other locations from location *i*.

For a second-order Markov chain model, the transition probabilities $\{p_{ijl}\}$ can be estimated by

$$\hat{p}_{ijl}(k) = \frac{\sum_{h=1}^{k} n_{ijl}(h)}{\sum_{s=1}^{N_Q}\sum_{h=1}^{k} n_{ijs}(h)} \tag{14}$$

where

$$n_{ijl}(k) = \begin{cases} 1 & \text{if state } i \text{ at time } k-2 \text{ and state } j \text{ at time } k-1 \text{ transitions to state } l \text{ at time } k \\ 0 & \text{otherwise} \end{cases}$$

We note that the Markov chain model considered here does not indicate the time duration when a pop-up threat remains visible. This time duration is also random. Hence, we need another procedure to estimate the time duration of appearance at a particular location (or cell). Let $t_{e\sup}(i,r)$ represent the estimated average appearance time duration of a pop-up threat stored at location *i*, and *r* denotes the last discrete event index when a pop-up threat actually appeared at location *i*. Let $t_{up}(i,r+1)$ denote the actual time duration of the threat appearing in cell i at the next discrete time event *r+1* it appears in location *i*. Each time a pop-up threat appears in location *i*, the estimated average appearance time duration is re-calculated, and it is given by:

$$t_{e\sup}(i,r+1) = \frac{t_{e\sup}(i,r)N(i,r) + t_{up}(i,r+1)}{N(i,r)+1} \tag{15}$$

where *N(i,r)* is the number of times the pop-up threat has appeared in location *i* up to time *r*. Note that in (13) and (14), the index *k* refers to the discrete event time when a pop-up threat appears somewhere,

whereas in (15) the index r refers to the previous discrete time when a pop-up threat appeared in location i. For example, when a pop-up threat appears in location i, having previously appeared at location s, all transition probabilities p_{sj} are updated for all $j \in Q$, using (13) for a first-order Markov model, but the average time duration update in (15) is performed only for location i.

B. Moving Horizon Estimates of Changing Transition Probabilities

As a military operation progresses, an adversary could possibly change the deployment strategies of pop-up threats. Such changes could result in variations of either the one-step transition probabilities and/or the order of the Markov chain model. Since it is not known whether the adversary has changed its deployment strategies, it would be inappropriate to use all the historical data to estimate the transition probabilities. Instead of using all data up to the present as given in expression (13) or (14), we remove possibly outdated data. This idea can be quantified via a moving-horizon approach. The transition probabilities $\{p_{ij}\}$ for the first-order Markov chain model are adaptively calculated as:

$$\hat{p}_{ij}(k) = \begin{cases} \dfrac{\sum_{h=k-H+1}^{k} n_{ij}(h)}{\sum_{s=1}^{N_Q} \sum_{h=k-H+1}^{k} n_{is}(h)} & \text{if } k \geq H \\ \dfrac{\sum_{h=1}^{k} n_{ij}(h)}{\sum_{s=1}^{N_Q} \sum_{h=1}^{k} n_{is}(h)} & \text{if } k < H \end{cases} \tag{16}$$

where H is the length of data used for estimation. Clearly, if the value of H is large enough, this estimation procedure is similar to that given in (13). The choice of the window H as a design parameter reflects previous experience or data on how long the transition probabilities remain constant before changes are made. On the other hand, when a threshold change is detected, one may then reduce H so that old data could be discounted sooner. Similarly, for the second-order Markov chain model, the one-step transition probabilities $\{p_{ijl}\}$ are estimated as

$$\hat{p}_{ijl}(k)=\begin{cases}\dfrac{\sum_{h=k-H+1}^{k} n_{ijl}(h)}{\sum_{s=1}^{N_Q}\sum_{h=k-H+1}^{k} n_{ijs}(h)} & \text{if } k \geq H \\ \dfrac{\sum_{h=1}^{k} n_{ijl}(h)}{\sum_{s=1}^{N_Q}\sum_{h=1}^{k} n_{ijs}(h)} & \text{if } k < H \end{cases} \qquad (17)$$

Next, we present an example to illustrate the performance of moving-horizon estimation. Consider an example of a first-order Markov chain with a given state space $Q=\{1,2\}$ and three different sets of one-step transition probability matrices $P_1^{(1)}$, $P_2^{(1)}$ and $P_3^{(1)}$ which are given by

$$P_1^{(1)}=\begin{bmatrix}0.5 & 0.5\\ 0.25 & 0.75\end{bmatrix},\ P_2^{(1)}=\begin{bmatrix}1/3 & 2/3\\ 0.5 & 0.5\end{bmatrix} \text{ and } P_3^{(1)}=\begin{bmatrix}0.75 & 0.25\\ 0.4 & 0.6\end{bmatrix}.$$

We use the procedure introduced in part C of Section II to create a sequence of cells with a length of 10,000 which is divided into three equal periods, and the sequence for the i^{th} period is associated with the transition probability matrix $P_i^{(1)}$ $(i=1,2,3)$. We applied the moving-horizon approach as expressed in (16) with $H=1000$ to estimate the varying transition probability $\hat{p}_{12}$. For comparison, we also applied the approach given in (13) using all the outdated data to estimate $\hat{p}_{12}$. The outcome is shown in Figure 2. We note that the estimated $\hat{p}_{12}$ using all the outdated data substantively deviates from the true value of corresponding transition probabilities, but the estimated $\hat{p}_{12}$ using the moving-horizon approach tracks the pop-up threat data generated by simulation more closely and more quickly.

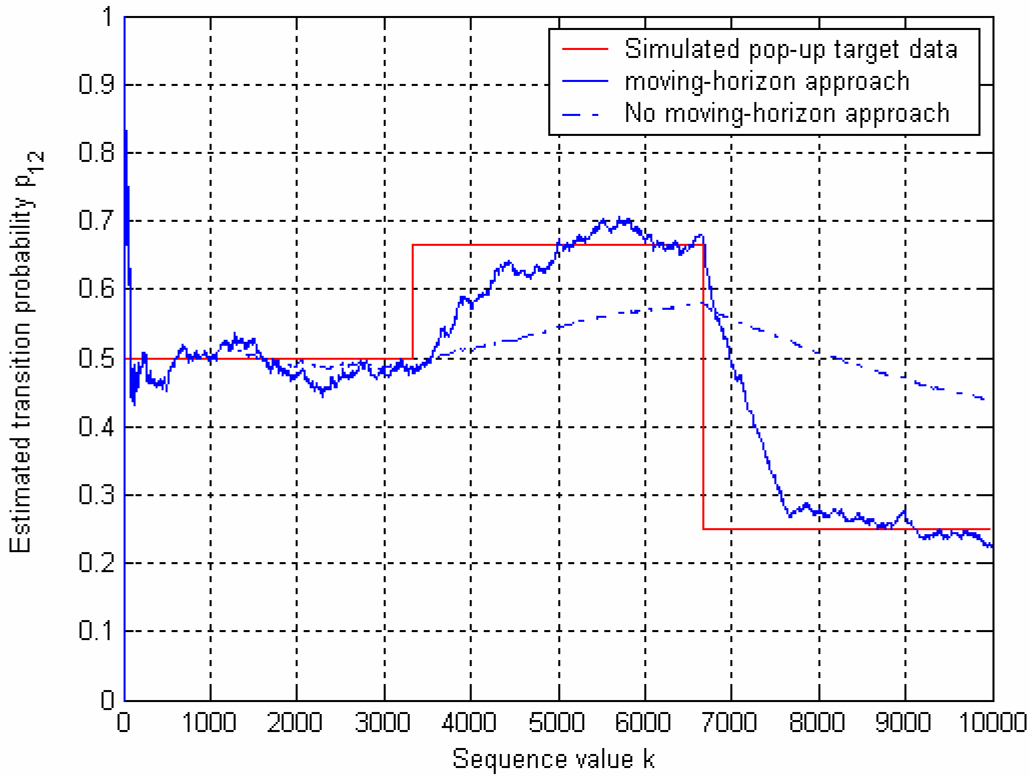

Figure 2. Comparison of various estimation approaches with the true value

C. Test for Choosing the Order of the Markov Chain Model

It should be noted that the approaches developed in this paper can be used to calculate the one-step transition probabilities only if the order of the Markov chain is reasonably chosen with respect to existing historical data. We need to choose the order of the model before we estimate the transition probabilities. In this section, we apply statistical inference to test the order [7]. The following hypothesis-testing algorithm is introduced to estimate the order of Markov chains with stationary transition probability matrices.

<u>*Step 1*</u>. Let $r=1$. The null hypothesis that the Markov chain model is of zero-order (independent process) versus the alternative that it is of order one or higher is given by

$$\mathrm{H}_0^{(r)}: p_{ij} = p_j \quad (i, j = 1, 2, \cdots, N_Q) \tag{18}$$

Let Λ_1 be a likelihood ratio test statistic corresponding to (18). It is

$$\Lambda_1 = 2\sum_{i=1}^{N_Q}\sum_{j=1}^{N_Q} n_{ij} \ln \frac{\hat{p}_{ij}}{\hat{p}_j} \tag{19}$$

where $\hat{p}_j$ is calculated as

$$\hat{p}_j = \frac{\sum_{l=1}^{N_Q} n_{jl}}{\sum_{j=1}^{N_Q}\sum_{l=1}^{N_Q} n_{jl}} \tag{20}$$

The test statistic Λ_1 has a chi-squared (χ^2) distribution with $N_Q^{r-1}(N_Q-1)^2 - d_r$ degrees of freedom where d_r is the number of zeros among $\left\{\hat{p}_j\right\}$. If the test statistic $\Lambda_1 \geq \chi^2_\alpha\left(N_Q^{r-1}(N_Q-1)^2 - d_r\right)$, we reject $\mathrm{H}_0^{(1)}$ under the significance level α (i.e., the process is not independent), and go to Step 2. Otherwise, we decide that the process is of zero-order and stop the test.

Step 2. Let $r = 2$. The null hypothesis that the Markov chain model is first-order versus the alternative that it is of order two or higher is given by

$$\mathrm{H}_0^{(r)}: p_{ijl} = p_{jl} \quad (i, j, l = 1, 2, \cdots, N_Q) \tag{21}$$

Let Λ_2 be a likelihood ratio test statistic corresponding to (21) which is calculated as

$$\Lambda_2 = 2\sum_{i=1}^{N_Q}\sum_{j=1}^{N_Q}\sum_{l=1}^{N_Q} n_{ijl} \ln \frac{\hat{p}_{ijl}}{\hat{p}_{jl}} \tag{22}$$

The test statistic Λ_2 has a chi-squared (χ^2) distribution with $N_Q^{r-1}(N_Q-1)^2 - d_r$ degrees of freedom where d_r is the number of zeros among $\left\{\hat{p}_{jl}\right\}$. If the test statistic $\Lambda_2 \geq \chi^2_\alpha\left(N_Q^{r-1}(N_Q-1)^2 - d_r\right)$, we reject $\mathrm{H}_0^{(2)}$ under the significance level α (i.e., the process is not first order), and go to Step 3. Otherwise, we decide that the process is of first-order and stop the test.

Step 3. Let $r = 3$. The null hypothesis that the Markov chain model is second-order versus the alternative that it is of order three or higher is given by

$$\mathrm{H}_0^{(r)}: \ p_{ijls} = p_{jls} \quad (i, j, l, s = 1, 2, \cdots, N_Q) \tag{23}$$

Let Λ_3 be a likelihood ratio test statistic corresponding to (23) which is calculated as

$$\Lambda_3 = 2\sum_{i=1}^{N_Q}\sum_{j=1}^{N_Q}\sum_{l=1}^{N_Q}\sum_{s=1}^{N_Q} n_{ijls} \ln \frac{\hat{p}_{ijls}}{\hat{p}_{jls}} \tag{24}$$

The test statistic Λ_3 has a chi-squared (χ^2) distribution with $N_Q^{r-1}(N_Q-1)^2 - d_r$ degrees of freedom where d_r is the number of zeros among $\left\{ \hat{p}_{jls} \right\}$. If $\Lambda_3 \geq \chi_\alpha^2\left(N_Q^{r-1}(N_Q-1)^2 - d_r \right)$, we reject $\mathrm{H}_0^{(3)}$ under the significance level α (i.e., the process is of order higher than two), and go to Step 4. Otherwise, we decide that the process is of second-order and stop the test.

Step 4. $\cdots$ (Continue testing)

We note that the value of $N_Q^{r-1}(N_Q-1)^2 - d_r$ becomes very large if the system has a very large state space size N_Q and a large value of r. However, the chi-squared (χ^2)-tables are generally not available for values of $\chi_\alpha^2\left(N_Q^{r-1}(N_Q-1)^2 - d_r \right)$ with a large degree of freedom. If the number of degrees of freedom is greater than 30, the χ^2 law can be approximated by a normal law [8]. Let $n = N_Q^{r-1}(N_Q-1)^2 - d_r$. If n is sufficiently large, the random variable $\sqrt{2\chi^2} - \sqrt{2n}$ is approximately normal. That is,

$$\Pr\left(a < \sqrt{2\chi^2} - \sqrt{2n} < b \right) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{s^2}{2}} ds \,. \tag{25}$$

With some re-arrangement, we have

$$\Pr\left(\chi^2 \leq \frac{(b+\sqrt{2n})^2}{2} \right) = \frac{1}{\sqrt{2\pi}} \int_{-\sqrt{2n}}^b e^{-\frac{s^2}{2}} ds = 1 - \alpha \,. \tag{26}$$

Given the value of significance α ($0 < \alpha \leq 1$), we can determine a value for b. Thus, we can determine the value of $\chi_\alpha^2\left(N_Q^{r-1}(N_Q-1)^2 - d_r \right)$ as given by

$$\chi_\alpha^2\left(N_Q^{r-1}(N_Q-1)^2-d_r\right)=\frac{\left(b+\sqrt{2(N_Q^{r-1}(N_Q-1)^2-d_r)}\right)^2}{2} \tag{27}$$

such that $\Pr\left(\chi^2 \geq \chi_\alpha^2\left(N_Q^{r-1}(N_Q-1)^2-d_r\right)\right)=\alpha$. Furthermore, if $N_Q^{r-1}(N_Q-1)^2-d_r \leq b$, we have $\chi_\alpha^2\left(N_Q^{r-1}(N_Q-1)^2-d_r\right)\leq N_Q^{r-1}(N_Q-1)^2-d_r$.

As mentioned earlier, changes in deployment strategy of pop-up threats may affect the order of the corresponding Markov chain process. Thus, we propose using a moving horizon hypothesis test which is a procedure similar to the one described in this section except that the hypothesis tests are calculated using a time-window of relatively recent data. This procedure is illustrated in Figure 3. Suppose the data is first generated from a first order Markov chain model, and then at time index k_1 the data is generated from a second order Markov chain model. In other words, during $k_0 \leq k < k_1$ the data is from a first-order model and during $k_1 \leq k \leq k_2$ the data is from a second-order model. At any time k_e we may estimate the order of the data using the hypothesis test with a moving horizon H. The data in the test is from the first-order model when $k_e \leq k_1$, or from the second-order model when $k_e \geq k_1 + H$, or from a mixture of first order and second order data when $k_1 \leq k_e \leq k_1 + H$.

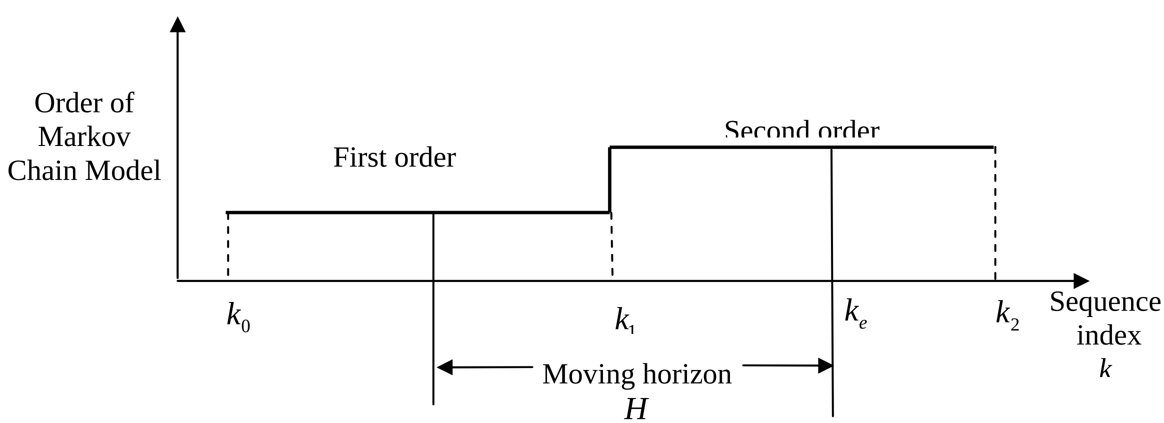

Figure 3. Moving horizon hypothesis test for the order of Markov chains

Next, we present an example to illustrate the performance of the moving-horizon order test. We use the procedure introduced in part C of Section II to create a sequence of cells with a length of 500 which is divided into two equal periods (i.e., $k_0 = 1$, $k_1 = 250$ and $k_2 = 500$), and the sequence for the first period is associated with a transition probability matrix P_1 and the sequence for the second period with a transition probability matrix P_2. Consider an example of a Markov chain with a given state space $Q = \{1, 2\}$ and two transition probabilities matrices which are given by

$$P_1 = \begin{bmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{bmatrix}, \text{ and } P_2 = \begin{bmatrix} 1/4 & 3/4 \\ 1/2 & 1/2 \\ 1/3 & 2/3 \\ 1/3 & 2/3 \end{bmatrix}$$

We applied the moving-horizon approach as explained before with $H = 50$ to test the order of the data. The result is shown in Figure 4. We observe that in the beginning the estimated order oscillates around the correct order. This is because the data points in the hypothesis test are not enough. As we use more data, the data points settle to the correct one (either the first order or the second order). We also note that during the transition period, the test results are not stable because the data used for the test includes points from both the first order and the second order processes.

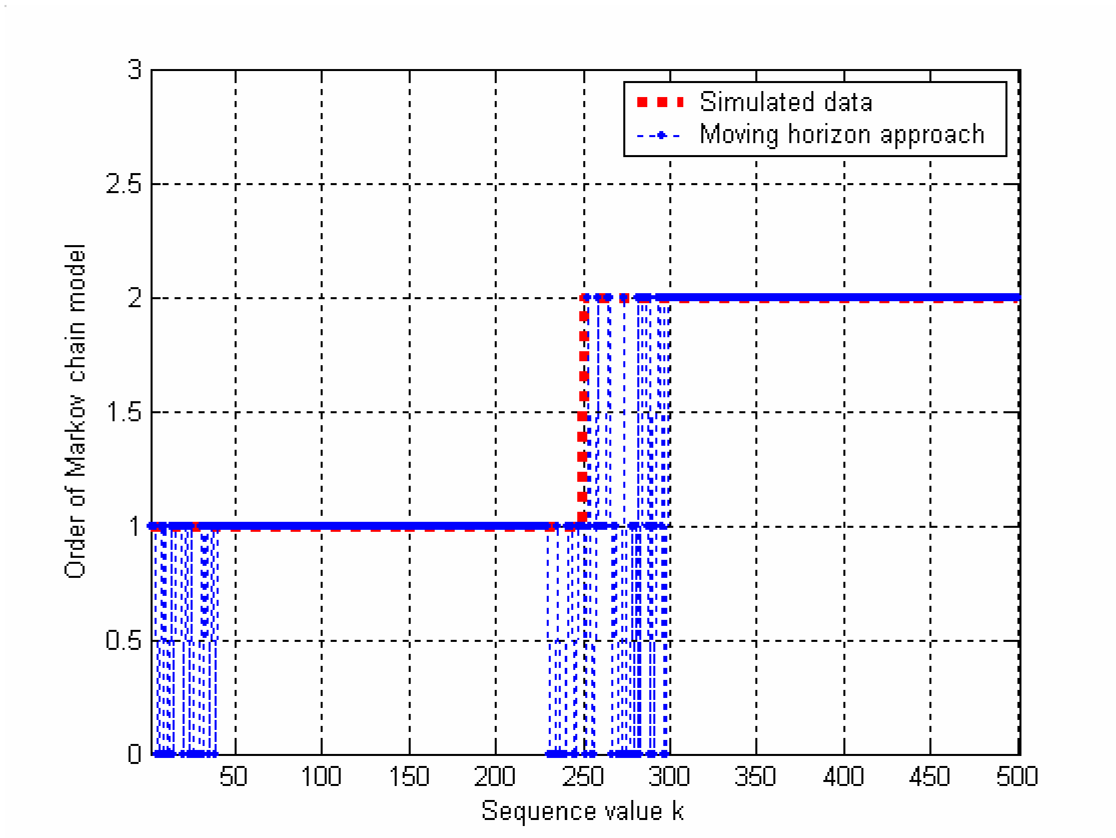

Figure 4. Moving horizon hypothesis order test

IV. COORDINATING NETWORKED UAVS FOR PAD MISSIONS

One of several generic military operations related to pop-up threats is a Persistent Area Denial (PAD) mission carried out by a group of UAVs.

A. Networked Unmanned Air Vehicles

Suppose that there are N_v UAVs and that the i^{th} UAV is modeled as a Dubins' car [9] and flies at a constant altitude and at a constant velocity, which has a continuous time kinematics given by

$$\begin{cases} \dot{x}_{v1}^i = v\cos\theta_v^i \\ \dot{x}_{v2}^i = v\sin\theta_v^i \\ \dot{\theta}_v^i = \omega_{\max} u_v^i \end{cases} \tag{28}$$

where x_{v1}^{i} is its horizontal position, x_{v2}^{i} is its vertical position, v is its transitional (constant) velocity, θ_v^i is its heading direction, $\omega_{\max}$ is its maximum angular velocity, and $-1 \le u_v^i \le 1$ is the steering input. Hence, $u_v^i = +1$ stands for the maximum turn to the right rate command, and similarly $u_v^i = -1$ represents the maximum possible left turn rate command. The minimum turn radius for the vehicles is $R = v / \omega_{\max}$. Rather than use this continuous time representation we assume that vehicles will travel either at the minimum turn radius or on straight lines. It is then possible to develop the analytical formulas for the vehicle trajectories (e.g. in terms of arc segments on circles and line segments [9]). We quantize these trajectories with a sampling interval T to obtain discrete time sequences that we denote by $x_{v1}^{i}(k)$, $x_{v2}^{i}(k)$, $\theta_v^i(k)$ and $u_i(k)$ for $k = 0,1,2,\ldots$. For convenience, let the state of the i^{th} UAV be

$$x_v^i = \left(x_{v1}^i, x_{v2}^i, \theta_v^i \right)'.$$

All UAVs are part of a communication network and assumed to have perfect location knowledge of threats that have popped up. The specific information that is shared among the UAVs includes the following: the current locations of UAVs, the current estimated transition probability matrix of the Markov-chain model, and agreement on which UAV attacks which pop-up threat. Delay and communication errors are not considered in this paper. Clearly, if these are modeled, the estimated transitional probability matrix held by different UAVs may differ. It is clear that the above assumptions on the communication capabilities of a group of UAVs will influence how well the cooperation strategy works [10].

B. Maneuvering UAVs to Pop-up Threats

To aid in the development of cooperative control strategies among UAVs, we address the issue of how to maneuver a UAV to an appropriate location prior to the appearance of pop-up threats. In the absence of having information about predicted pop-up threat locations, a UAV may wait for the next

actual pop-up threat to appear and then maneuvers to that threat. We call this type of strategy Waiting (WA) locating strategy. On the other hand, a UAV may decide to move to an intermediate position before the real pop-up threat appears. This may be done if predicted information is available. We studied several strategies of locating a UAV to a temporary position in [1-3]. In [1-2], we only considered the maximum likelihood (ML) locating strategy while, in [3], we investigated additional locating strategies such as the Bayesian weighted (BW) locating strategy and uniformly weighted (UW) locating strategy. In most cases [3], it was shown that the ML locating strategy works best in terms of minimizing the expected time to reach the next most likely pop-up threat after it appears. Thus, we only consider the ML locating strategy in this paper.

Consider a present pop-up threat at cell i, and the corresponding next pop-up threats predicted from the Markov chain model as illustrated in Figure 5. Let S denote the set of cells where those predicted pop-up threats could be located;

$$S = \{i_1, i_2, \cdots, i_s\},$$

where transition probability $p_{ii_j} > 0$ for all $j = 1, \cdots, s$ and s denotes the number of possible cells.

Moreover, let $x^{i_j} = [x_1^{i_j}, x_2^{i_j}]^T$ be the coordinate of the j^{th} predicted pop-up threat. We now define the maximum likelihood locating strategy to redirect a UAV to an intermediate position based on the predicted information.

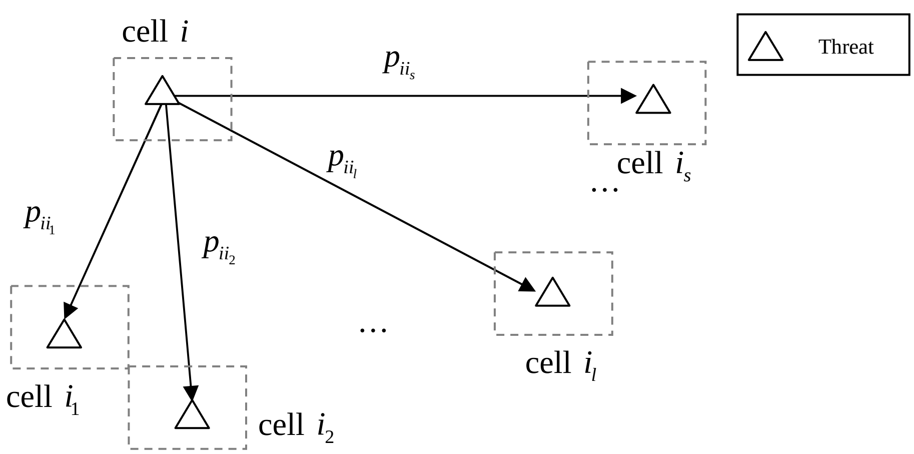

Figure 5. Present threat at cell i and the corresponding predicted pop-up threats in the set S

<u>*ML Strategy.*</u> A predicted location $x^{ML} = [x_1^{ML}, x_2^{ML}]^T$ selected as

$$x^{ML} = x^{i_j^*} \text{ with } i_j^* = \arg\max_{i_j}\{p_{ii_j}\}$$

is called the maximum likelihood (ML) locating strategy.

Using this strategy, a UAV will choose the targeting location at the predicted pop-up threat with maximum transition probability.

C. Cooperative Control Strategies of Networked UAVs

It is desirable to have more than one UAV performing persistent area denial since multiple UAVs could cooperate by sharing relevant information and they could reduce pop-up threat prosecution time. A cooperative strategy among multiple UAVs has been presented in [2]. A comparison of the results using non-cooperative and cooperative methods shows that the cooperative control improves the overall mission effectiveness by accomplishing the PAD operation in the shortest time and addressing more pop-up threats. In the study, the information about predicted pop-up threats, however, was not utilized in the cooperative assignment of multiple UAVs. As a result, the UAVs waited for the next pop-up threat to appear even after they have completed their current tasks. Ideally, as explained in section B, it would be appropriate if UAVs could continually re-optimize their operation by taking into account the locations of the predicted appearance of pop-up threats, and preposition themselves to respond. In this section, we define a specific version of our cooperative control strategy by modifying the one developed in [3]. The principal difference is that the information about predicted pop-up threats is used in the cooperative strategies. The algorithm is given as follows:

1) Initial Cooperative Target Assignment: Consider the allocation of multiple UAVs to the threats that are currently observed. The cost function and constraints are formulated as

$$\min_{\psi(i,j)}\left\{\sum_{j=1}^{N_r(k)}\sum_{i=1}^{N_v(k)} d(x_v^i, x_r^j)\Omega(i,j)\right\} \tag{29}$$

where $N_v(k)$ is the number of the UAVs available at time *k* and $N_r(k)$ is the number of the threats observed at time *k*, subject to the constraints

$$\sum_{j=1}^{N_r(k)} \Omega(i, j) = 1 \quad \text{for any } i = 1, \cdots, N_v(k) \tag{30}$$

$$\sum_{i=1}^{N_v(k)} \Omega(i, j) = 1 \quad \text{for any } j = 1, \cdots, N_r(k) \tag{31}$$

where $\Omega(i, j)$ is the assignment function which is given by

$$\Omega(i, j) = \begin{cases} 1 & \text{if UAV } i \text{ is assigned to target } j \\ 0 & \text{otherwise} \end{cases}. \tag{32}$$

In the cost function, the expression $d(x_v^i, x_r^j)$ is the approximate flying distance from the current location of UAV i to the pop-up threat j, ignoring the z coordinates for simplicity where x_v^i and x_r^j represent the x-y coordinates of UAV *i* and threat *j*, respectively. The constraints (30) and (31) mean that each UAV can be assigned to a single threat only and vice versa. Note that the variable $\Omega(\cdot,\cdot)$ can take values of 0 or 1 only. The best assignment $\left\{\psi^*(i, j), i = 1, \cdots, N_v(k); j = 1, \cdots, N_r(k)\right\}$ is obtained via minimizing the cost function over all possible assignments $\left\{\psi(i, j), i = 1, \cdots, N_v(k); j = 1, \cdots, N_r(k)\right\}$. This can be done by using an integer programming (IP) approach [11]. For the purpose of illustration, we consider two UAVs, i.e., $N_v(k) = 2$ for all *k*. Let AV (UV) and AT (UT) denote the set of assigned (unassigned) UAVs and assigned (unassigned) pop-up threats, respectively, which represent the outcomes of this algorithm. Let Dim(X) denote the dimension of the set X.

2) Cooperative Targeting Adjustment: In solving (29), we do not consider the distance between the pop-up threats in the set of assigned threats. It is possible that in certain situations, a single UAV is sufficient to destroy two or more targets close to each other. In addition, the allocation of UAVs to

targets depends on the remaining expected time of the visibility of the targets, i.e. $\{t_{e\sup}(i), i \in Q\}$. The adjustment of the initial allocation is as follows. Consider the situation where $AT = \{j_1, j_2\}$ and $Dim(AT) = 2$. This means that the two UAVs are assigned to attack the two targets. We consider that two targets are close to each other if the distance between them is less than a threshold d_{th}. In this case, one UAV can be assigned to attack both of them sequentially. To determine which one will be assigned, we define a distance index $dist(i, j_1 \to j_2)$ which represents the distance that the UAV i is required to travel on a path that arrives at target j_1 and then target j_2. This distance can be calculated as

$$dist(i, j_1 \to j_2) = d(i, j_1) + d(j_1, j_2).$$

The UAV i^* to be assigned is determined as

$$i^* = \arg\min_{i_s} \begin{Bmatrix} dist(i_s, j_1 \to j_2), \\ dist(i_s, j_2 \to j_1), i_s = 1, 2 \end{Bmatrix}.$$

and UAV i^* can reach both targets before they are expected to disappear. This algorithm makes an adjustment based on the initial target assignment, thus we call it an adjustment algorithm. If the set of unassigned targets is not empty, then the other UAV $i' \neq i^*$ will be assigned to a target in the set UT closest to it. If $Dim(UT) \geq 2$, UAV i' may determine how many targets it can deal with using the adjustment algorithm. If $UT = \varnothing$, the set of assigned UAVs (AV) and the set of unassigned UAVs (UV) are updated as :

$$AV = AV / \{UAV_{i'}\} \text{ and } UV = UV \bigcup \{UAV_{i'}\}.$$

3) Predicted Target Assignment: When there is no target unassigned, the unassigned UAVs will determine an intermediate location to move to by using the locating strategies developed in part B based on the predicted information obtained from Markov model, instead of waiting for the next pop-

up target to appear. If there are more than two unassigned UAVs, the UAV which is closest to the intermediate location will be redirected towards the predicted target.

D. Monte Carlo Simulation and Analysis

To implement and evaluate cooperative strategies, we used a Matlab simulation first developed at AFRL [12] which was extended by the authors as a test bed. We considered the area of PAD operation in Figure 1. We assumed that, for each current pop-up location, the next pop-up location is not identical to the current one. UAVs were assumed to move at a constant speed of $v = 40m/\sec$. Then, for a given initial location $(x_i(k), y_i(k))$ of a UAV and its heading direction $\theta_i(k)$, and a location of a pop-up threat and its heading angle, the code will generate trajectories or waypoints between these two locations, which are the minimum time/distance trajectories. To demonstrate the utility of the Markov chain model for predicting pop-up threats and the effectiveness of developed strategies for cooperative UAVs, two different Monte Carlo simulation tests are presented for a four-UAV team. In the first simulation, we illustrate the importance and influence of order test on mission performance. In the second simulation, we investigate the performance of cooperative strategies when the transition probabilities are changing. In both simulations, the locations of threats are generated from the Markov chain model using the procedure introduced in Section II. We assume that the estimated appearance time of pop-up threats at each cell has been obtained using the adaptive approach before the UAVs start their PAD mission.

We use the method of Section C to generate test data for constructing a Markov chain model of order one and order two. Thus from the test data we construct the initial estimated transition probability matrix of the model. Then we used the same method of Section C to generate more test data. This second set of data is used in conjunction with the model to predict the location of the next pop-up threat. In addition, the second set of data is used to update the estimated transition probability matrix of the model. In potential applications, the test data would be replaced by real data.

Simulation 1:

In this analysis, the locations of the pop-up threats are generated from a second-order Markov chain with stationary transition probabilities. At first, the simulation is performed by implementing the cooperative strategies with mixed information of predicted and actual pop-up threats where the transition probabilities are estimated for a first-order Markov chain model. We then perform the same simulation with the transition probabilities calculated for a second-order Markov chain model. Three values of appearance time for the pop-up threats considered are used: 30 seconds, 40 seconds and 50 seconds. For each set, we generated 10 samples of sequences of the pop-up location. The performance index is defined as the average percentage of pop-up threats reached over the 10 samples. Figure 6 shows the results. We evaluated the performance using a second order model and also using a first order model. The results are shown in Figure 6, which indicates that a second order model has a better mission performance than a first order model when test data is generated from a second-order model as well. It is desirable to use a first order model for simplicity and minimum computation compared to higher order. Using the order test method proposed in part C of Section III with enough data points available, we could determine that the actual data is from a second order model and so a second order model would be a good model.

In the next example, we observe the performance of a first order model and a second order model when the test data is from a first order process. We generated the location of pop-up threats from a first-order Markov chain with stationary transition probabilities. We considered a first-order model and a second-order model when we implemented the developed cooperative strategies for a UAV team. Comparing the results of these two situations in Figure 7, we note that there is no advantage in using a second-order model, and the results are even worse under certain conditions. We note that, for a first-order test data, we have

$$\Pr\left(Z_{k+1}=l \mid Z_k = j, Z_{k-1}=i\right)=\Pr\left(Z_{k+1}=l \mid Z_k = j\right), \qquad i, j, l \in Q \tag{33}$$

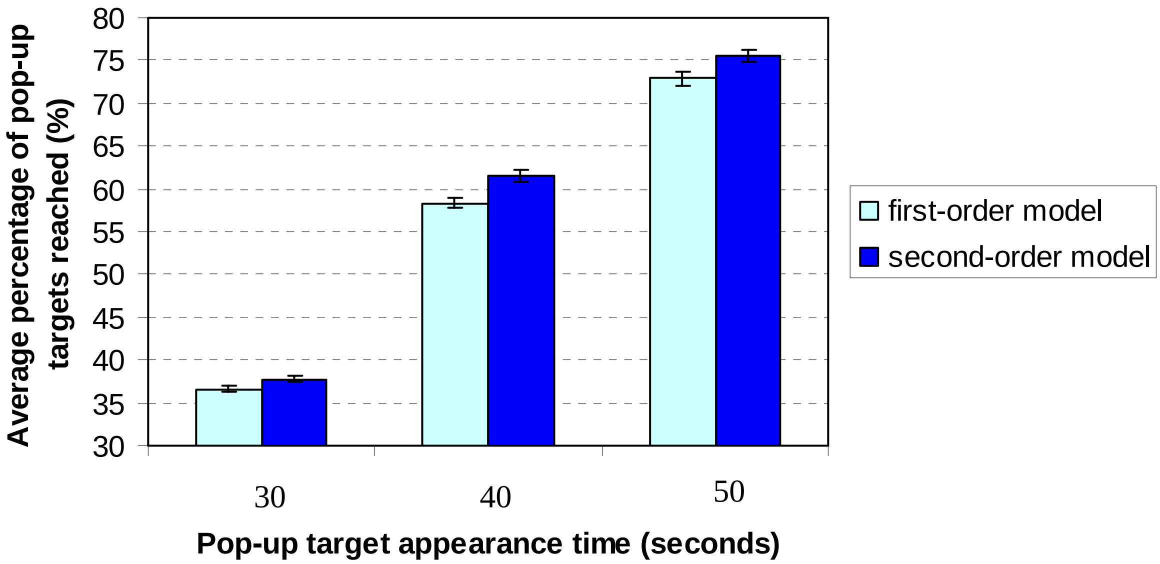

Figure 6. Comparison outcome with second-order Markov test data.

(The standard deviations for the averages are indicated by short bars on the graphs.)

That is, $p_{ijl} = p_{jl}$ for any $i, j, l \in Q$. In other words, there is no difference for maneuvering UAVs based on either first-order or second-order models. In that sense, we can see that the results shown in Figure 7 are reasonable. Moreover, the overestimation of the order in this situation does not enhance the mission performance and also increases the computational effort. We conclude that it is prudent not to overestimate the process model.

Simulation 2 (Piece-wise stationary transition probabilities):

In this analysis, the appearance of pop-up locations is generated from a first-order Markov chain process. The transition probabilities used to generate those data, however, are changed during the scenario progress (piece-wise stationary). Two sets of appearance times for the pop-up threats are considered: 30 seconds and 35 seconds. For each set, we generated 10 samples of sequences for the

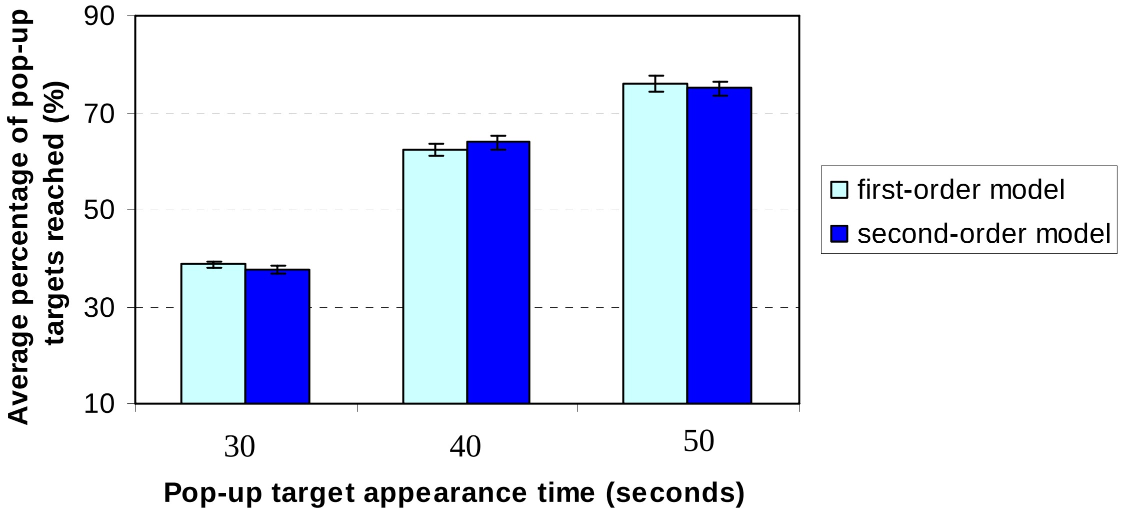

Figure 7. Comparison outcome with data from a first-order Markov chain process

(The standard deviations for the averages are indicated by short bars on the graphs.)

pop-up locations. The performance index is defined as the average time the UAV teams (over 10 samples) are required to reach 10 pop-up threats after the transition probabilities are changed. This measure represents the speed with which various transition-probability estimation approaches can respond to the change. This index is similar to the response rise time to a step input in standard control theory. The results with and without a moving horizon approach are shown in Table 1. We observe that, in both cases, utilizing the moving-horizon approach has a faster response, and thus improves the performance of cooperative UAVs.

In both simulations we demonstrated the importance of using a moving horizon *H*, and choosing an appropriate value so that in the event that the transition probabilities associated with a data sequence change (piece-wise stationary), we can adapt our model to the change in a timely manner. When we choose *H* to be very short, we can respond to changes quickly but our averaging accuracy would be degraded. If there is an intentional change in the order representing the data sequence (due to changes

in adversarial strategies), a moving horizon for the order test would permit an adaptive switch to a different model order, so the predictions for future locations of pop-up threats would be more accurate.

Table 1. Averaged response time for various estimation approaches.

(Standard deviations are given in parenthesis.)

Appearance time	30 seconds (σ)	35 seconds (σ)
No moving-horizon approach	1071.2 seconds ($\pm$10.6s)	668.2 seconds ($\pm$9.8s)
Moving-horizon approach with $H = 30$	627.8 seconds ($\pm$14.5s)	474.9 seconds ($\pm$17.1s)

V. CONCLUDING REMARKS

The randomness of pop-up threat appearance can be quantified via a Markov model. An n^{th}-order Markov chain models the dependence of the appearance of pop-up threats on previous appearances. We considered first-order and second-order Markov chain models with stationary or piece-wise stationary transition probabilities. A moving-horizon approach is proposed for estimating the changing transition probabilities in a timely manner. We also introduced a hypothesis testing method from statistical inference theory to determine an appropriate order of the Markov model. A PAD mission, one of typical military applications, carried out by multiple UAVs, is used to illustrate the effectiveness of Markov models for predicting pop-up threats. Cooperative strategies were designed to achieve the objectives of UAVs based on available information about predicted pop-up threats, thus enabling better results. Our simulation results demonstrate that the developed cooperative

strategies using combined information about pop-up threats can reduce mission time and increase the number of the threats reached, and thus improve the overall performance of PAD missions. In this paper, we used test data generated from given Markov chains, constructed first and second order models, and used similar data for performance evaluations. In the future, we will use data from representative real pop-up locations in constructing the Markov chain models and in evaluating their performance in representative PAD missions.

VI. REFERENCES

[1] Shankar K. Subramanian, Jose B. Cruz, Phillip R. Chandler and Meir Pachter, "Predicting Pop Up Threats from an Adaptive Markov Model, " Chapter 19 in *Recent Developments in Cooperative Control and Optimization*, Editors: Sergiy Butenko, Robert Murphey, and Panos Pardalos, Kluwer Academic Publishers, Norwell Massachusetts, 2003 , pp. 357-366.

[2] Shankar K. Subramanian and Jose B. Cruz, "Adaptive Models of Pop-Up Threats for Multi-Agent Persistent Area Denial", in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Hawaii, December, 2003, pp. 510-515.

[3] Yong Liu, Jose B. Cruz, Jr. and Andrew G. Sparks, "Coordinating Networked Uninhabited Air Vehicles for Persistent Area Denial," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, December, 2004.

[4] Jiecai Luo, "Some New Optimal Control Problems in UAV Cooperative Control with Information Flow Constraints," in *Proceedings of the American Control Conference*, Denver, Colorado, June 4-6, 2003, pp.2181-2186.

[5] A. Gil, S. Ganapathy, K. Passino, and A. Sparks, "Cooperative scheduling of tasks for networked autonomous vehicles," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Hawaii, 2003, pp.522-527.

[6] Olle Haggstrom, *Finite Markov Chains and Algorithmic Applications*, Cambridge University Press, 2002.

[7] U. Narayan Bhat, *Elements of Applied Stochastic Processes*, Second Edition, John Wiley& Sons, 1984.

[8] J. Bass, *Elements of Probability Theory*, Academic Press, 1966.

[9] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and targets," *American Journal of Mathematics*, Vol. 79, No. 3, July, 1957, pp.497-516.

[10] J. Finke, K. M. Passino, S. Ganapathy, and A. Sparks, "Modeling and analysis of cooperative control systems for uninhabited autonomous vehicles," in *Cooperative Control* (S. Morse, N. Leonard, and V. Kumar, eds.), Springer-Verlag, 2004.

[11] H. Fourer, D. M. Gay, and B.W. Kernighan, *AMPL, A modeling language for mathematical programming*, The Scientific Press, 1993.

[12] S. J. Rasmussen and P. R. Chandler, "MultiUAV: A multiple UAV simulation for investigation of cooperative control," in *Proceedings of the 2002 Winter Simulation Conference*, San Diego, CA, Vol.1, pp. 869-877.